Virtualization

(and cloud computing at CERN)

Ulrich Schwickerath

Special thanks: Sebastien Goasguen
Belmiro Moreira, Ewan Roche, Romain Wartel

See also related presentations:

- CloudViews2010 conference, Porto
- HEPIX spring and autumn meeting 2009, 2010
- Virtualization vision, Grid Deployment Board (GDB) 9/9/2009
- *Batch virtualization at CERN, EGEE09 conference, Barcelona*
- *1st and 2nd multi-threading and virtualization workshop, CERN, 2009*

The organizers wish list for this presentation

Questions to be addressed in this presentation

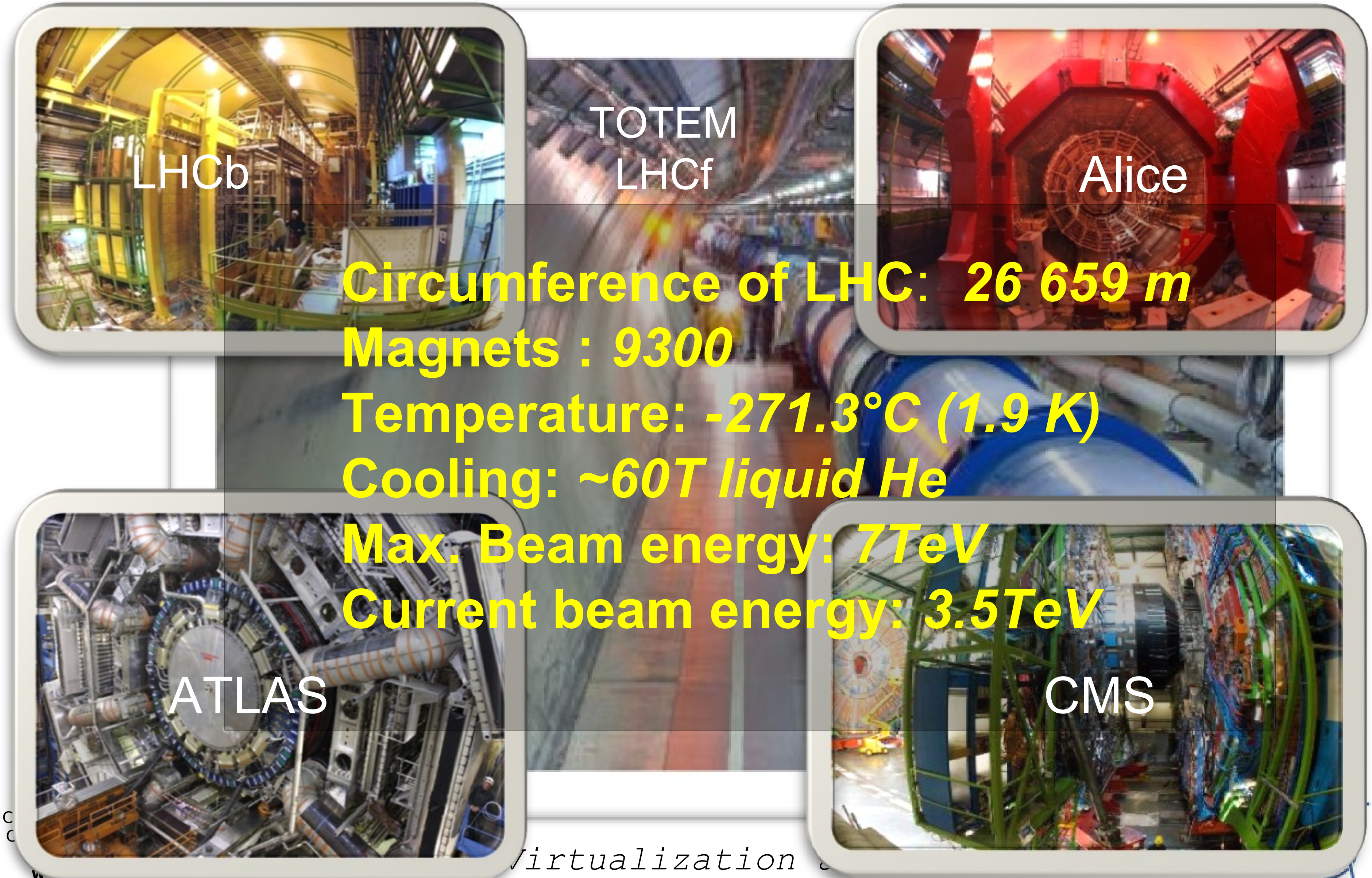
- what is the motivation to go for virtualization and why do we need it
- how does the technical implementation of an IaaS architecture at CERN look like
  - how to manage systems for large scale virtualization
  - how we do stuff like image distribution
- which experiences/lessons did we learn already

Some facts about CERN

European Organization for Nuclear Research

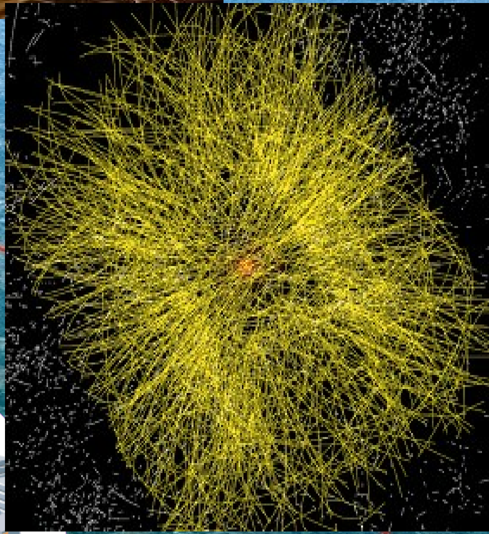
- **The world's largest particle physics laboratory**
- **Located on Swiss/French border**
- **Funded/staffed by 20 member states in 1954**
- **With many contributors in the USA**
- **Birth place of World Wide Web**
- **Made popular by the movie "Angels and Demons"**
- **Flag ship accelerator LHC**
- **http://www.cern.ch**

Facts about LHC



- **Circumference of LHC:** ***26 659 m***
- **Magnets :** ***9300***
- **Temperature:** ***-271.3°C (1.9 K)***
- **Cooling:** ***~60T liquid He***
- **Max. Beam energy:** ***7TeV***
- **Current beam energy:** ***3.5TeV***

Facts about LCG computing



Data Signal/Noise ratio 10^{-9}

Data volume:

High rate * large number of channels * 4 experiments

- ➔ **15 PetaBytes of new data each year**

Compute power

Event complexity * Nb. events * thousands users

- ➔ **100 k CPUs (cores)**
- ➔ **20% of this at CERN**

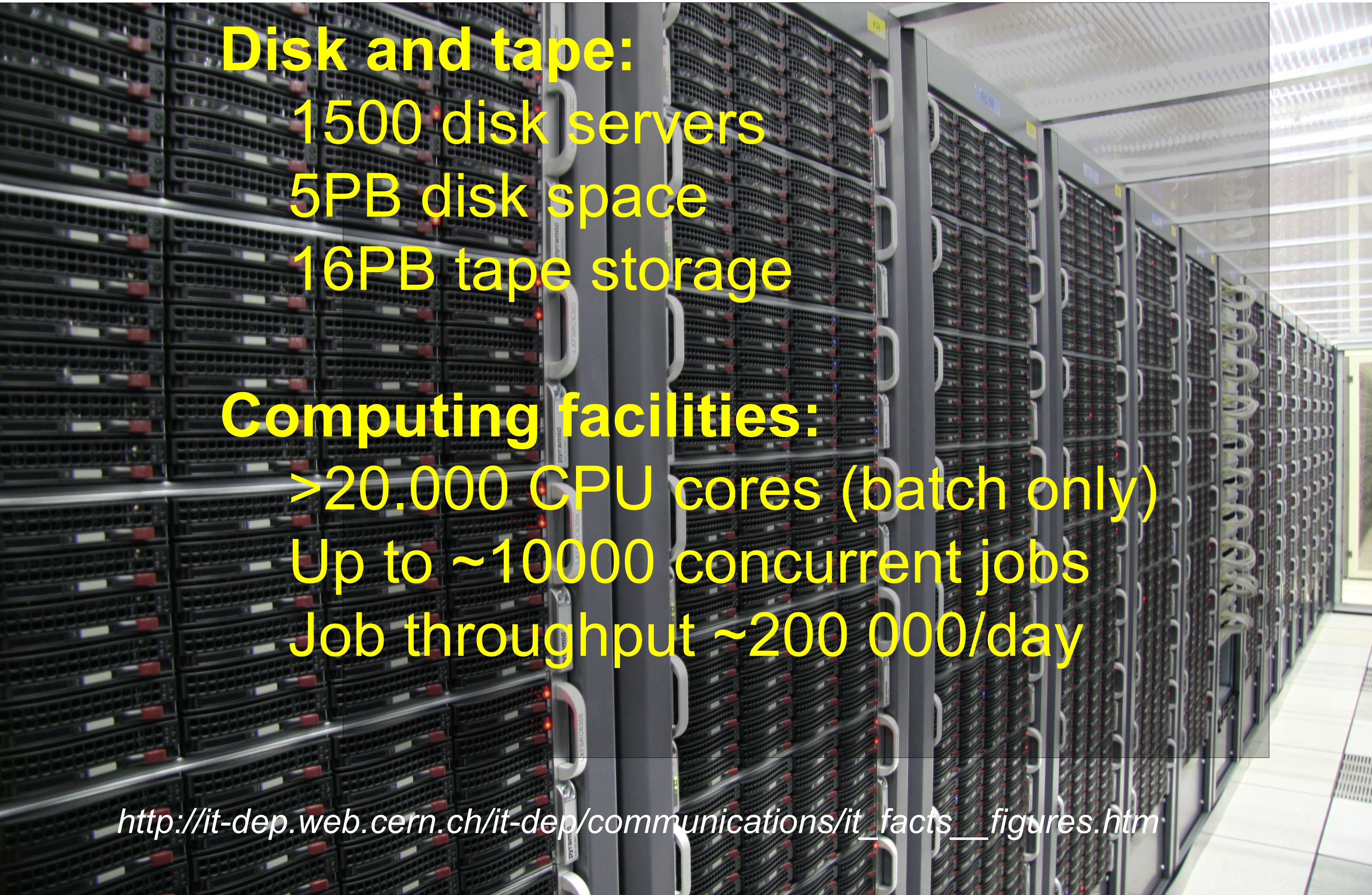
Worldwide analysis & funding

Computing funding locally in major regions & countries

Efficient analysis everywhere

- ➔ **GRID technology**

Facts about CERN Computer Center



Disk and tape:

1500 disk servers
5PB disk space
16PB tape storage

Computing facilities:

>20.000 CPU cores (batch only)
Up to ~10000 concurrent jobs
Job throughput ~200 000/day

http://it-dep.web.cern.ch/it-dep/communications/it_facts__figures.htm

The organizers wish list for this presentation

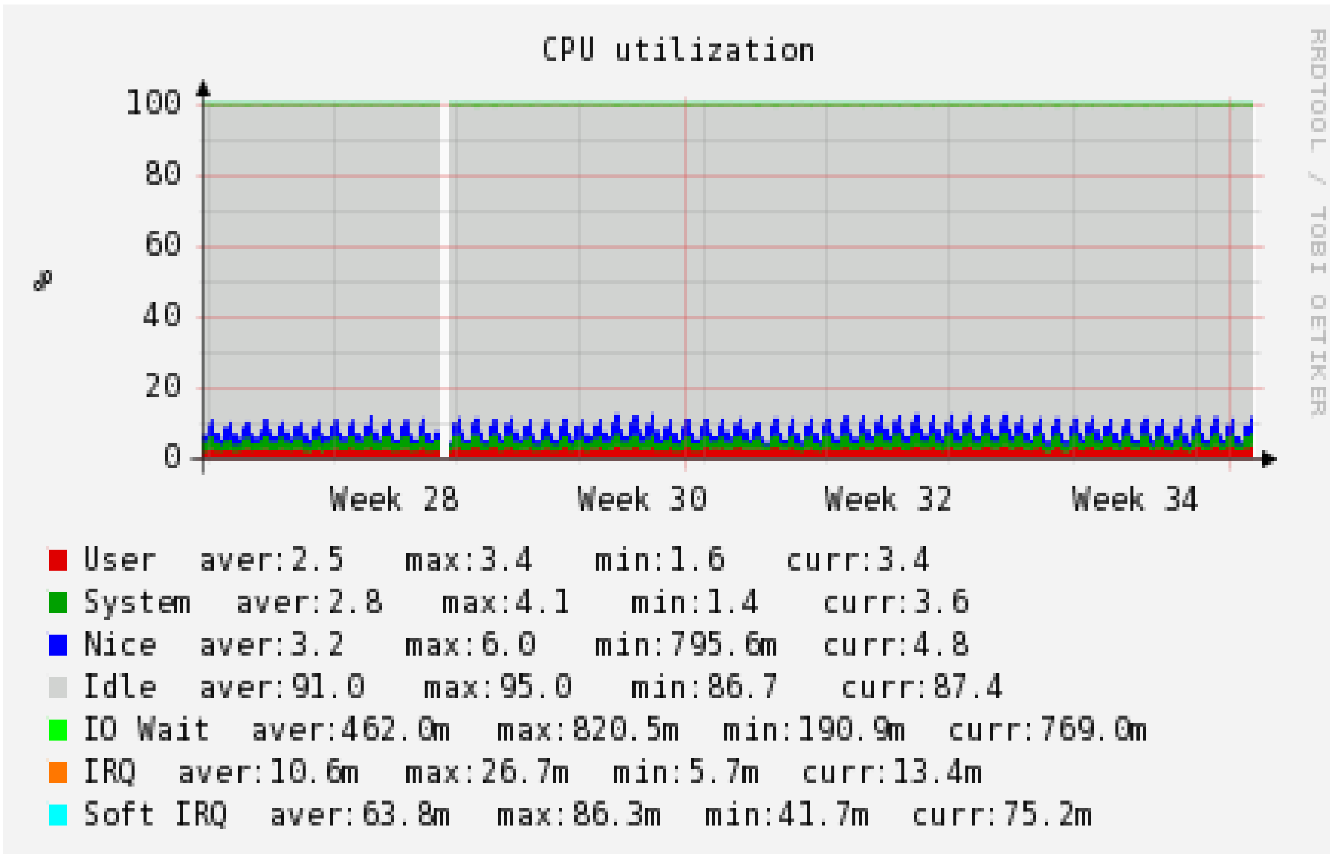
Questions to be addressed in this presentation

- what is the motivation to go for virtualization and why do we need it
- how does the technical implementation of an IaaS architecture at CERN look like
  - how to manage systems for large scale virtualization
  - how we do stuff like image distribution
- which experiences/lessons did we learn already

Example from CERN:

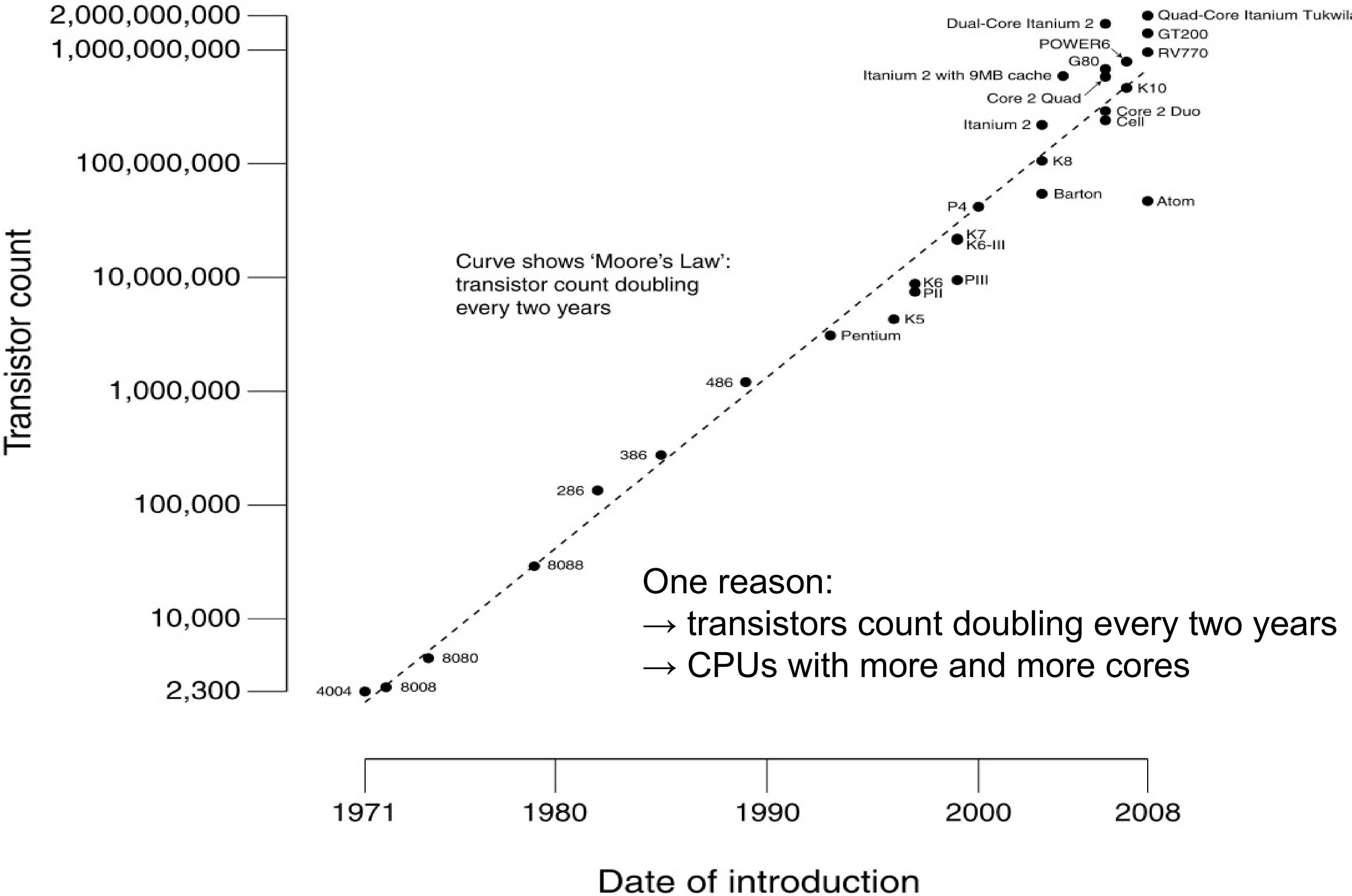
CPU utilization of dedicated resources for special tasks, eg web servers

155 machines



CERN IT Department
CH-1211 Genève 23
Switzerland
www.cern.ch/it

CPU Transistor Counts 1971-2008 & Moore's Law



One reason:

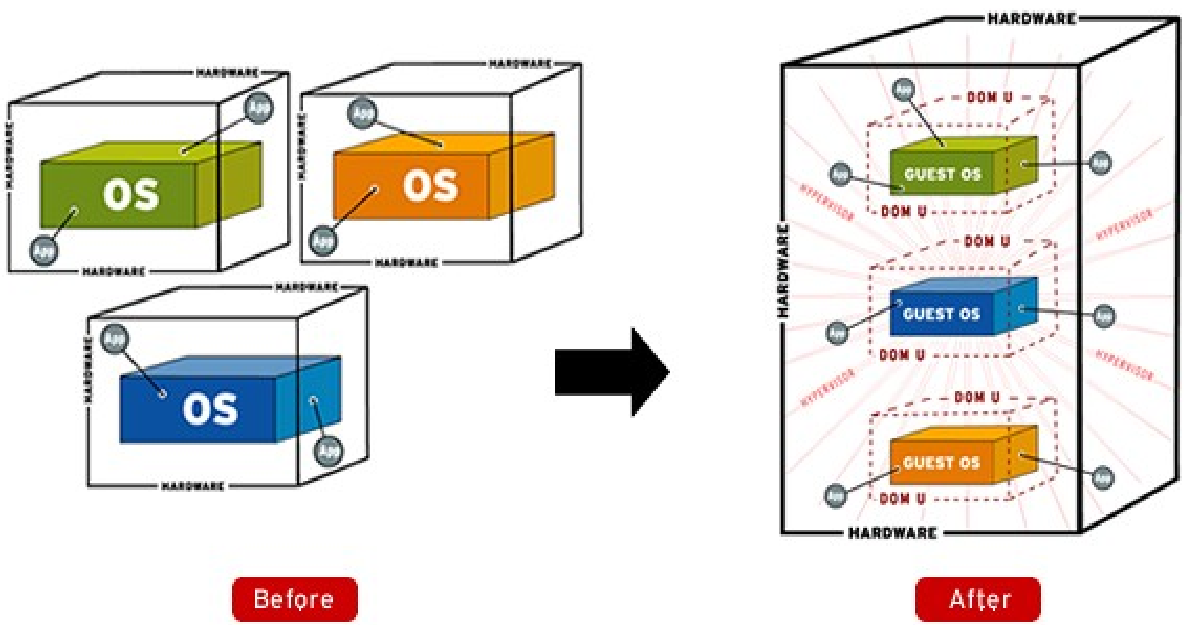
→ transistors count doubling every two years

→ CPUs with more and more cores

Virtualization: why ?

Service consolidation

- Improve resource usage by squeezing underutilized machines onto single hypervisors
- Ease management by supporting live migration
- Decouple hardware life cycle from applications running on the box

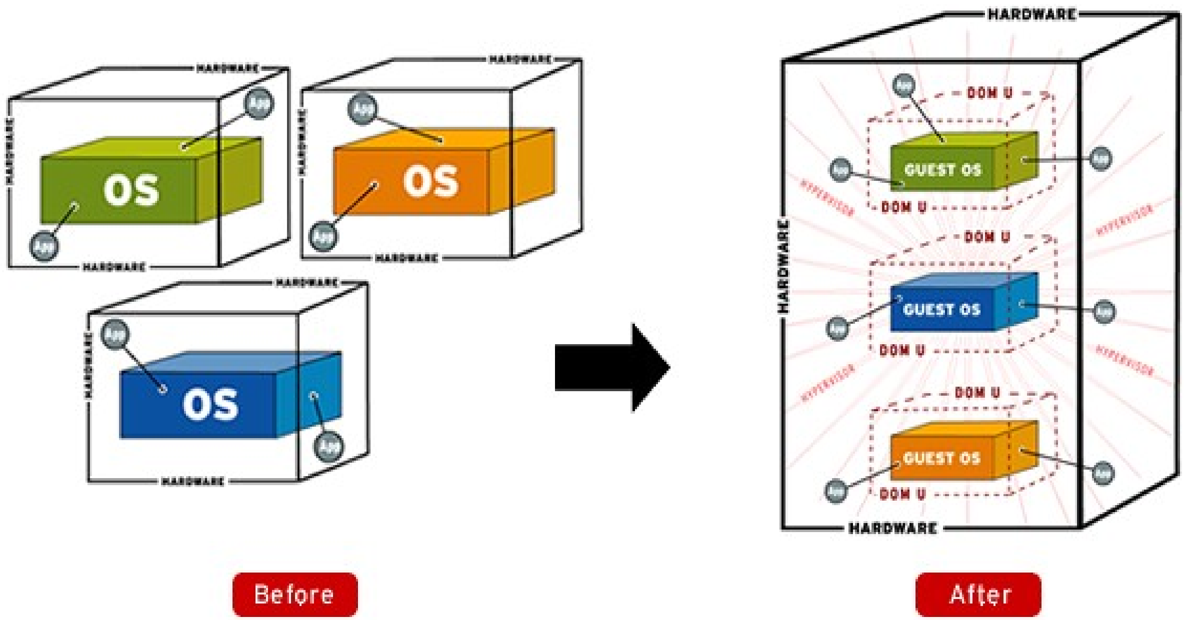


CERN IT Department
CH-1211 Genève 23
Switzerland
www.cern.ch/it

Virtualization: why ?

Service consolidation

- Improve resource usage by squeezing underutilized machines onto single hypervisors
- Ease management by supporting live migration
- Decouple hardware life cycle from applications running on the box

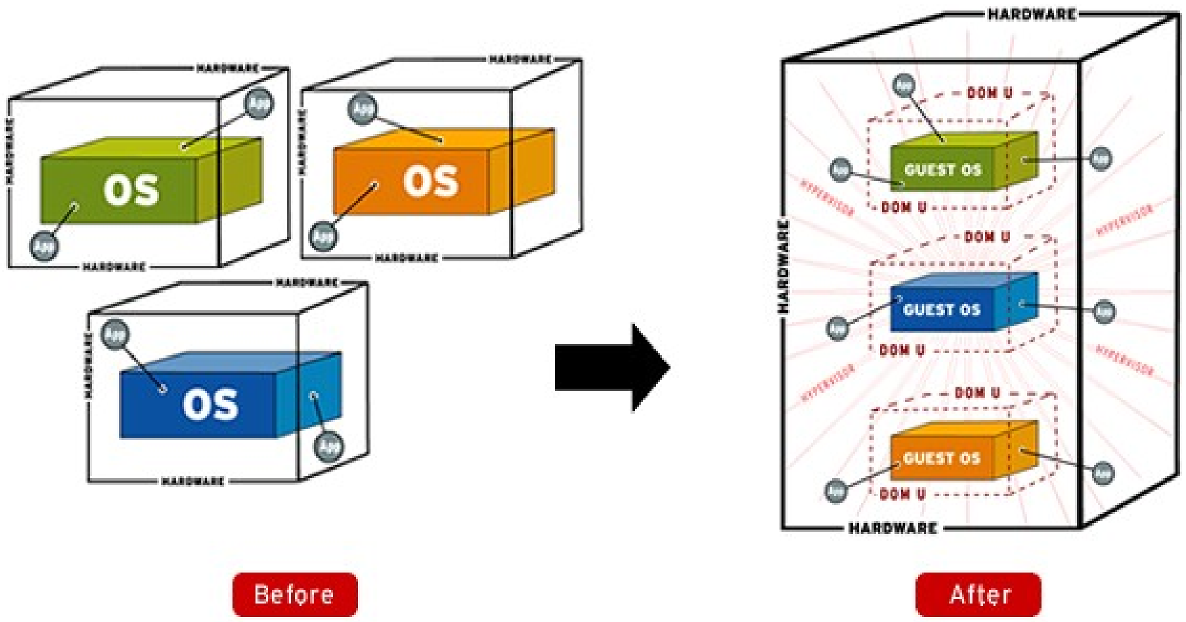


CERN IT Department
CH-1211 Genève 23
Switzerland
www.cern.ch/it

Virtualization: why ?

Service consolidation

- Improve resource usage by squeezing underutilized machines onto single hypervisors
- Ease management by supporting live migration
- Decouple hardware life cycle from applications running on the box



Virtualization: why ?

CERN IT Department

Service consolidation

- Improve resource usage by squeezing underutilized machines onto single hypervisors
- Ease management by supporting live migration
- Decouple hardware life cycle from applications running on the box

What does that mean ?

Virtualization: why ?

Hardware life cycle:

Hardware is replaced every 3 years:

- Hardware is bought with 3 years warranty
- New hardware is more power efficient

The dilemma:

- new hardware often requires new operating systems/drivers
- Old software often does not work on new operating systems ...

Virtualization can solve this problem !

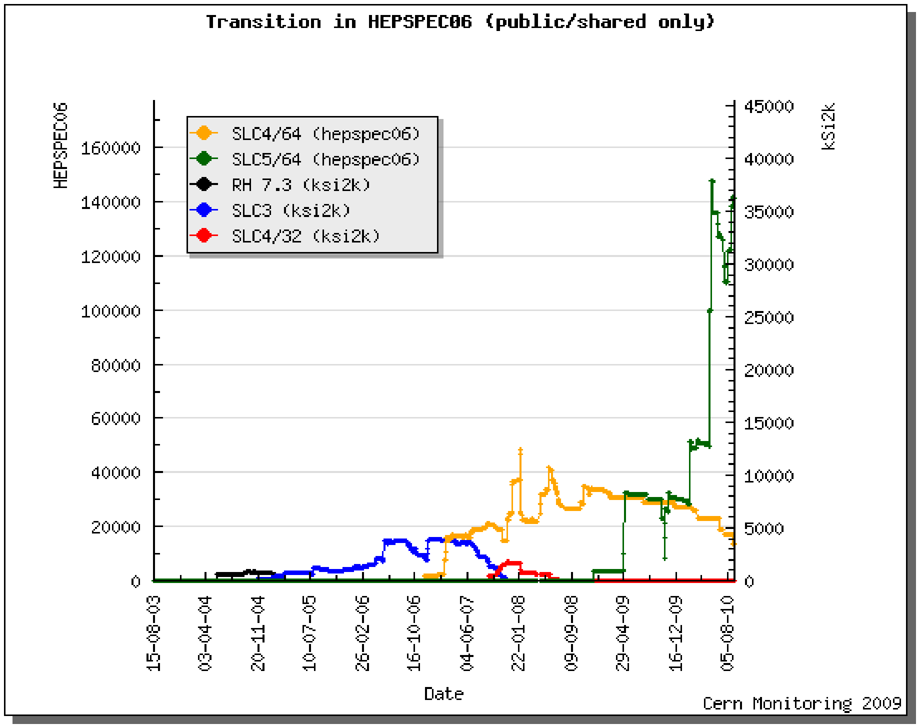
Note: Also operating systems have a limited life time !

Service consolidation at CERN: just now being put into production

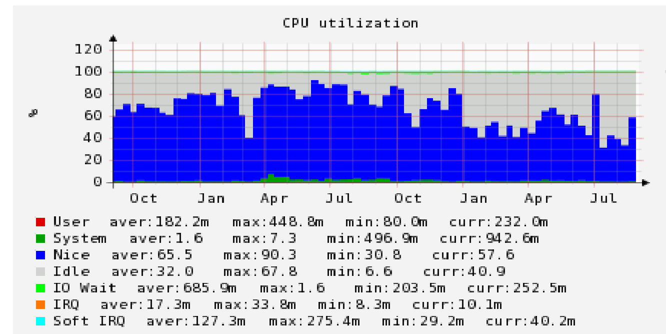
Virtualization for batch processing ?

Situation for the batch farm at CERN

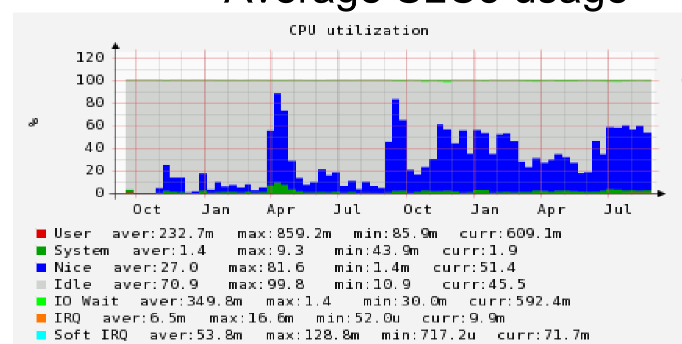
Issue: worker node transition to new OS versions



Average SLC4 usage



Average SLC5 usage



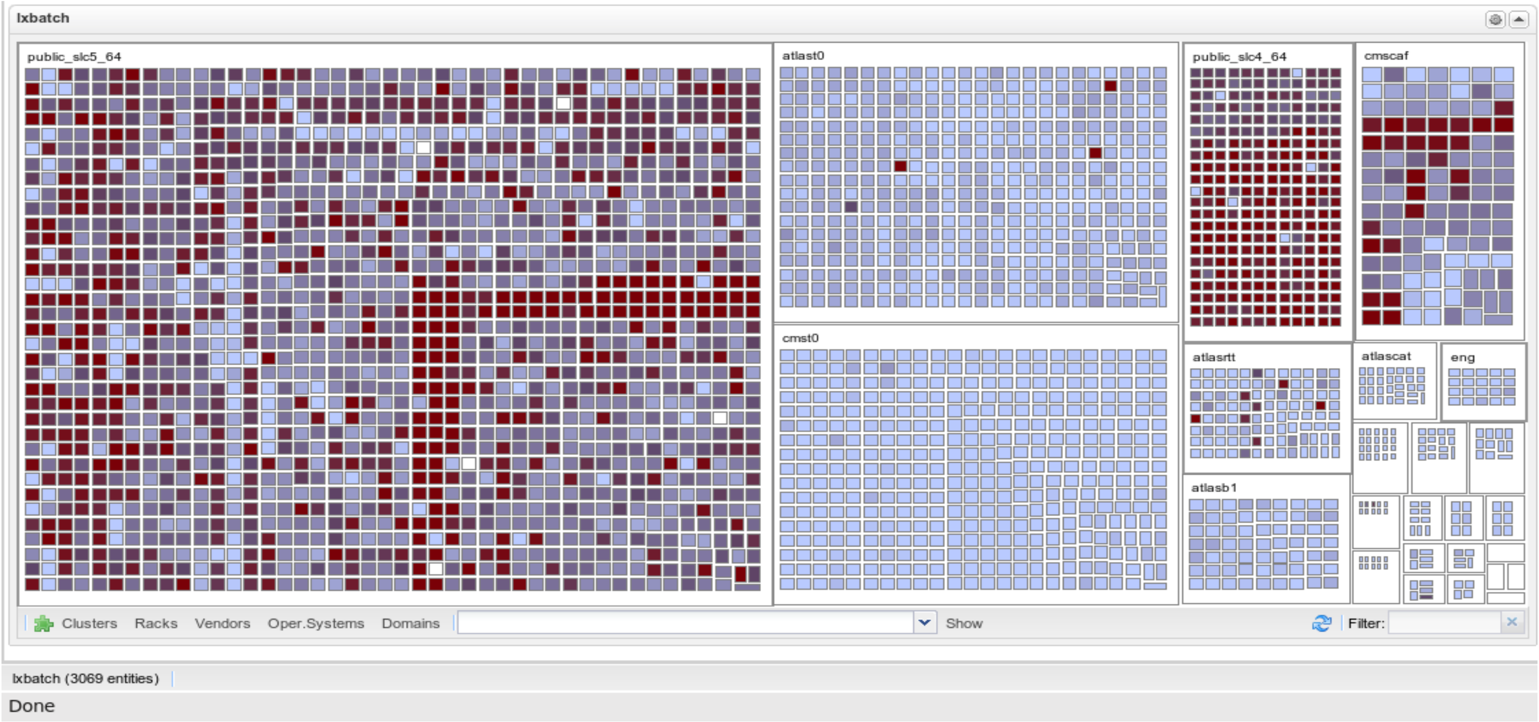
SLC4/SLC5 mixture:

- New resources didn't work with SLC4
- Not all applications were ready for SLC5

The batch farm at CERN: some facts

Issue: increasing complexity complicates scheduling of intrusive updates:

- ~ 80 different queues
- ~ 50 different node functions and sub-clusters
- ~ 40 different hardware types



CERN IT Department
CH-1211 Genève 23
Switzerland
www.cern.ch/it

Virtualization of lxbatch: Motivation

Facts: CERN batch farm lxbatch

~3500 physical hosts

~20000 CPU cores

>80 queues

Challenges:

- How to resolve conflicts between new hardware and old OS ?
- How to provide the right mix of environments matching needs ?
- How to manage intrusive interventions in a fragmented setup ?

Can virtualization help here ? YES!

Virtualizing batch resource

Challenges:

- How to resolve conflicts between new hardware and old OS ?
- How to provide the right mix of environments matching needs ?
- How to manage intrusive interventions in a fragmented setup ?

Idea:

Use virtual batch worker nodes!

You will need

- a **scalable and robus**t infrastructure which can support many thousand virtual machines
- a way to manage these virtual machines

Virtualizing batch resource

Challenges:

- How to resolve conflicts between new hardware and old OS ?
- How to provide the right mix of environments matching needs ?
- How to manage intrusive interventions in a fragmented setup ?

You will need

- to support different virtual machine images of all required flavors
- have an efficient way to **transport** these to your physical nodes
- a **decision making system** which
  - chooses the hardware
  - starts the virtual machine
  - monitors it during it's lifetime

Challenges:

- How to provide the right mix of environments matching needs ?
- How to resolve conflicts between new hardware and old OS ?
- How to manage intrusive interventions in a fragmented setup ?

Idea:

If you limit the life time of your virtual batch worker nodes, and always restart from the latest image version, you get your intrusive software upgrades for free !

For hardware intervention, just take the physical node out of production, and wait until all VMs are gone. That's equivalent to draining of physical batch nodes !

The organizers wish list for this presentation

Questions to be addressed in this presentation

- what is the motivation to go for virtualization and why do we need it
- how does the technical implementation of an IaaS architecture at CERN look like
  - how to manage systems for large scale virtualization
  - how we do stuff like image distribution
- which experiences/lessons did we learn already

Virtualizing batch resources ...

Cloud Computing

- SaaS (e.g Google docs), PaaS (e.g Google App Engine), IaaS (e.g EC2) layers
- IaaS layer offers Elasticity, On-demand and utility pricing
- IaaS enabled by virtualization

Glossary

SaaS : Software as a Service

PaaS : Platform as a Service

IaaS: Infrastructure as a Service

See presentation from Tony Cass !

Virtual Batch Principles

Virtual batch worker nodes:

- Clones of real worker nodes but not directly centrally managed
- Can coexist with physical worker nodes in lxbatch
- Instantiated proactively depending on demand
- **Dynamically join the batch farm** as worker nodes
- Limited lifetime: **stop accepting** jobs after 24h,

  and then gets destroyed when empty
- Only one user job per VM at a time

Note:
The limited lifetime allows for a fully automated system which dynamically adapts to the current needs, and facilitates the deployment of intrusive updates.

Virtual Batch Principles

Image creation

- Derived from a centrally managed "**golden node**"
  - Ensures that they are always up to date
- Stored in an image repository
- Pushed to the hypervisors in case of an update

Images

- Staged on hypervisors to speed up instance creation
- Master images, instances use LVM snapshots
- Start with few different flavours only (e.g SLC5, SLC4)

Virtual Batch: Principles

Image distribution

- Currently only shared file system available is AFS
- Scalability for O(1000) hypervisors is an issue
- **Resolved by using** peer to peer methods
  - SCP wave
  - Rtorrent
- Close collaboration with HEPiX virtualization Working Group

SCP wave

Principles:

- Binary tree based distribution
- Each node which finished the transfer becomes a new parent node
- Slow at the beginning, but with a logarithmic speed up

Available at:

http://www.opennebula.org/software:ecosystem:scp-wave

CERN IT Department
CH-1211 Genève 23
Switzerland
www.cern.ch/it

BitTorrent setup

- Several implementations were tested, rtorrent was finally selected
- Careful tuning is required:
  - The number of connected peers is restricted to at most 5
  - The memory usage for the transfers needs to be tuned
- We use a central tracker based on OpenTracker, complemented with DHT
- The tracker runs an rtorrent client and acts as initial sender

If you are interested, get in touch with us

Image distribution: other options

- NFS : discarded because of scalability concerns. It would require a dedicated infrastructure to make it scale to O(1000) physical machines
- AFS : even with replication performance and scalability concerns
- CERNVMFS : may be an option but requires several layers of squid proxies which we currently don't have (yet ?)
- Multicast: possibly the most efficient way, but currently not (yet?) supported by the CERN networking infrastructure

Virtual machine management system

The VM management system must perform the following tasks:

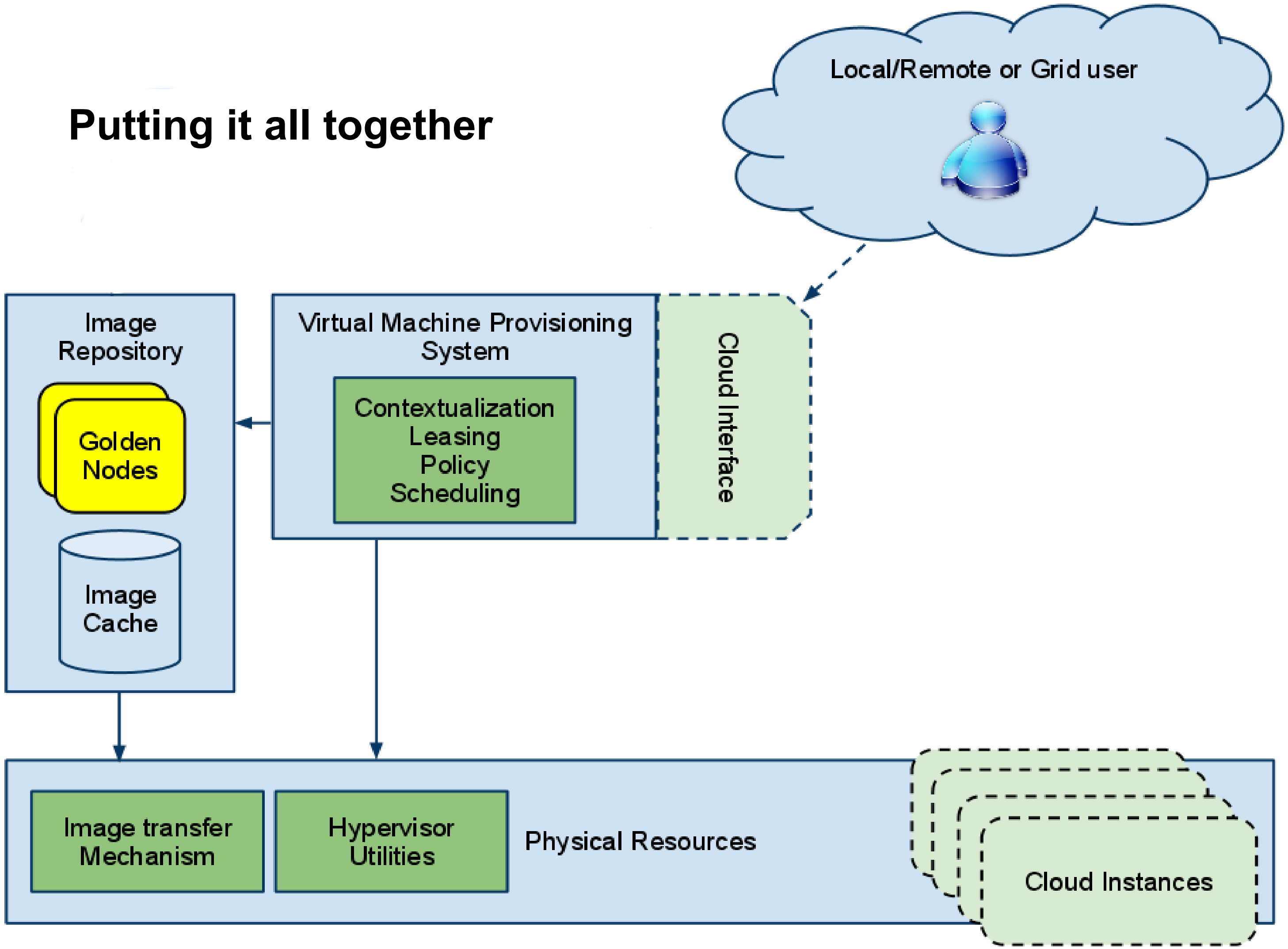
- Receive and process requests for virtual machines
  - Provide a queue for pending requests
  - Choose the next request to be processed
  - Terminate them when requested
- Choose a matching physical machine with enough free resources
- Talk to the selected machine, start the virtual machine and monitor it

VM placement and management system

- We count on existing solutions
- Tested both an open source and a proprietary solution
  - OpenNebula (ONE)
  - Platform's Infrastructure Sharing Facility (ISF)
- Both implement a request queue and a basic scheduler to place VMs

Virtual Batch: Architecture

Putting it all together



Status of building blocks (test system)

	Batch application	Hypervisor cluster	VM kiosk and image distribution	VM management system
Initial deployment	OK	OK	OK	OK
Central management	OK	OK	Mostly implemented	ISF OK, ONE missing
Monitoring and alarming	OK	(OK) Switched off for tests	missing	missing

Questions to be addressed in this presentation

- what is the motivation to go for virtualization and why do we need it
- how does the technical implementation of an IaaS architecture at CERN look like
  - how to manage systems for large scale virtualization
  - how we do stuff like image distribution
- which experiences/lessons did we learn (already)

Lessons learned

Most important: **scalability** and reliability at all levels !

- Hypervisor cluster
  - Manage it centrally like a big farm of unique machines
  - Never do manual interventions on the machines
- Image distribution system
  - It must be designed to scale to several thousand clients
  - The distribution time must be reasonable
- Virtual machine provisioning system
  - Must be able to deal with several thousand hypervisors
  - Must be able to manage several ten-thousand machines
  - Must allow for a redundant setup (eg fail-over)
- Batch system software
  - Must be able to manage many thousand clients

All of these need careful testing

Scalability tests: can we actually do it ?

Temporary access to about **500** new physical worker nodes

- 2 x XEON L5520 @ 2.27GHz CPUs (2 x 4 cores)
- 24GB RAM
- ~1TB local disk space
- HS06 rating ~97
- SLC5 **XEN** (1 rack with **KVM** for testing)
- Up to **44 VMs** per hypervisor (on private IPs)

Temporary batch master machine(s):

- 2 x XEON L5520 @ 2.27GHz CPUs (8 cores)
- 48GB RAM
- 3x 2TB disks, software raid 5 setup
- 10 GE

Scalability tests: hypervisor cluster

All machines in the hypervisor cluster

- Belong to the same cluster called **lxcloud**
- Share the same software setup
- Are centrally managed with the Quattor tool kit
- Are monitored with LEMON

CERN has several years of operational experience with such a setup. We know that it will scale up to several thousand machines.

New nodes automatically download images and sign up to ONE.

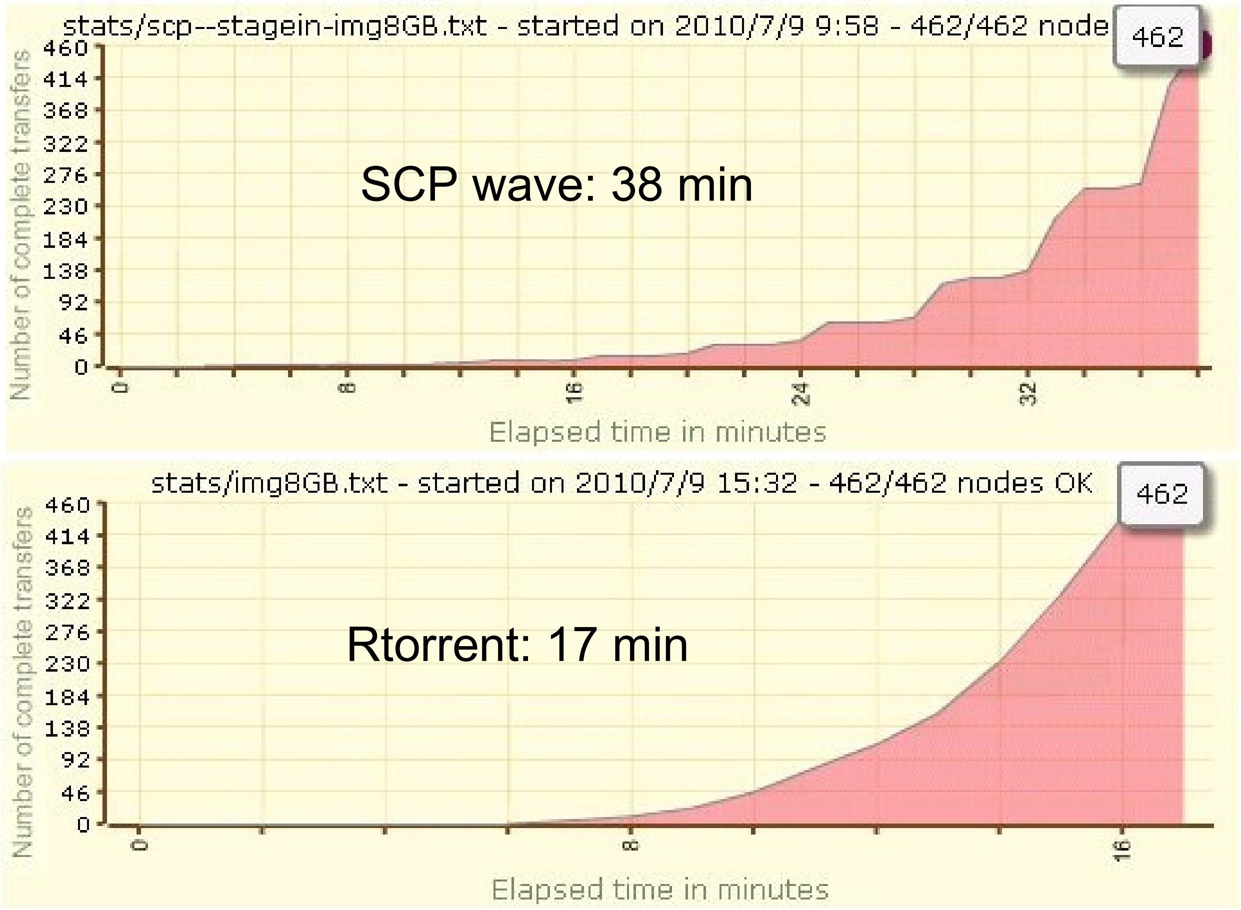
http://elfms.web.cern.ch/elfms

Scalability tests: Image distribution system

CERN IT Department

Image Distribution: 8GB test file to 462 hypervisors

Number of target machines



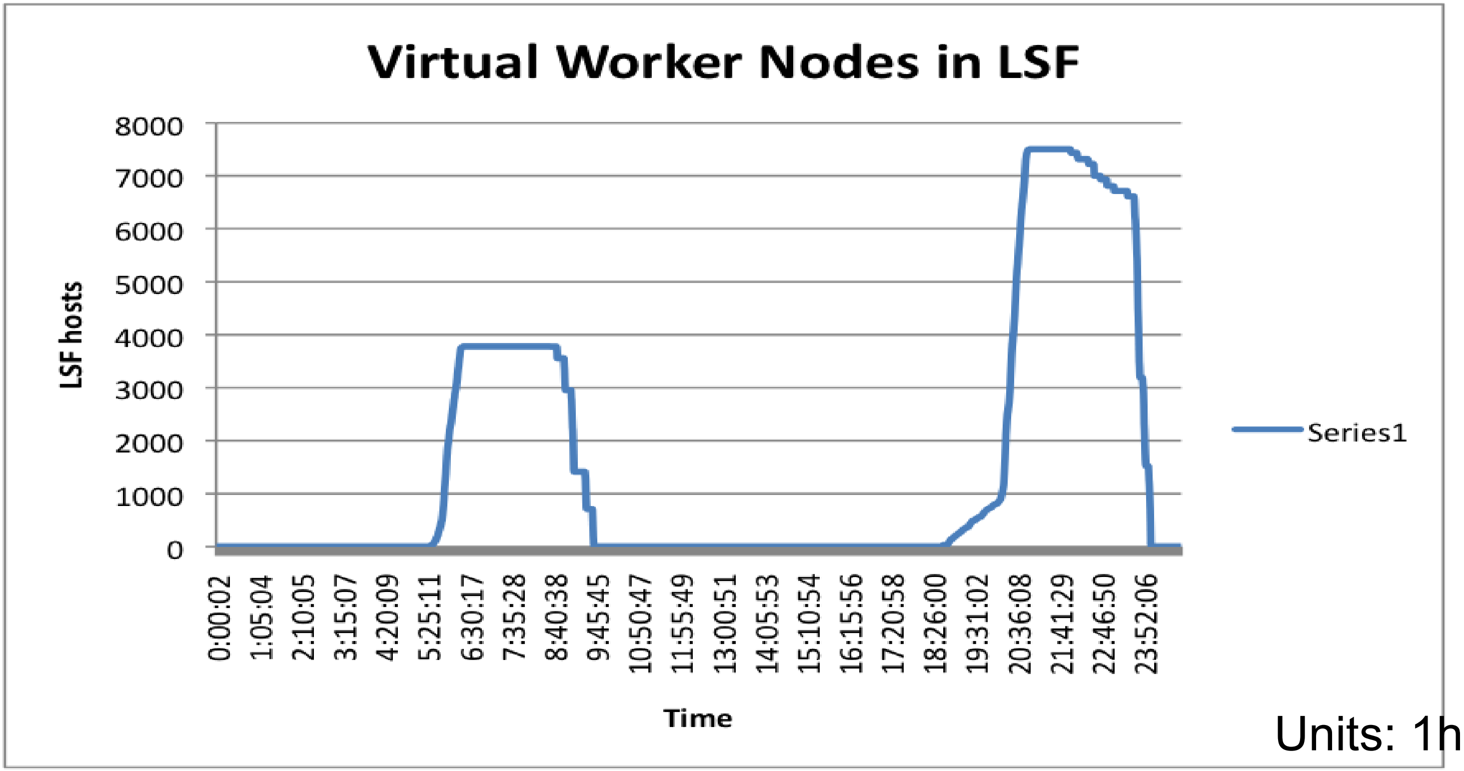
Throttling possible

time/min

Scalability tests: provisioning systems

"One shot" test with OpenNebula:

- Inject virtual machine requests
- And let them live and die, reduced life time 4h
- Record the number of alive machines seen by LSF every 30s



Units: 1h5min

Scalability tests: provisioning systems

Experiences:

- We started with a few hundred virtual machines
- Scaling up the systems required close collaboration with developers
- Both systems are capable to manage several thousand machines now

Scalability tests: batch system software

Questions to address

- Up to which number of worker nodes does LSF scale ?
- What size is required for the LSF master nodes ?
- What is the response time as a function of worker nodes ?
- At which point do we have to redesign the batch service ?

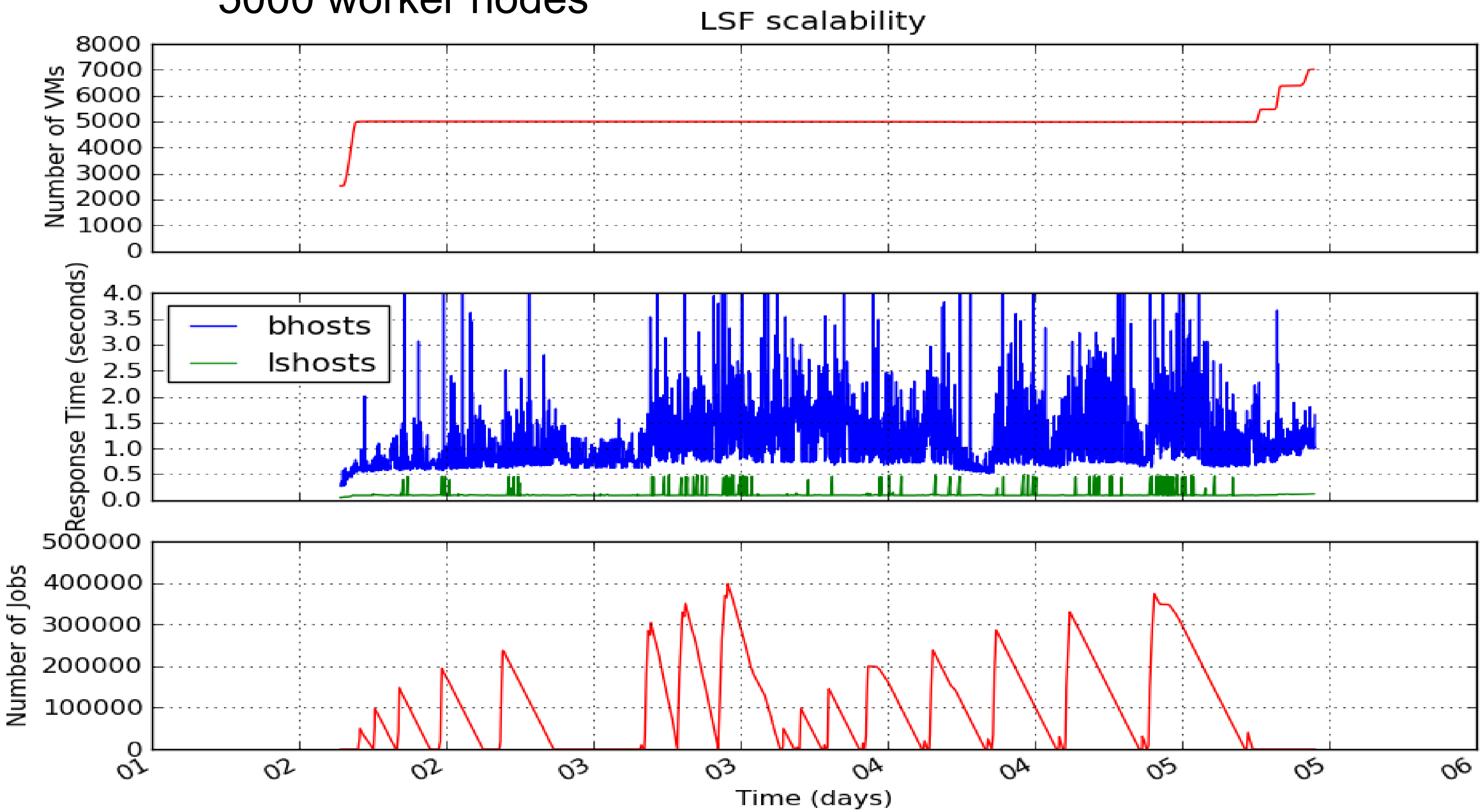
Note: These questions are independent of virtualization! The new infrastructure allowed us probe the software way beyond what can be done with physical resources ! This is of high interest for planning of the layout of computing resources.

Remark: LSF=Load Sharing Facility, provided Platform Computing

Scalability tests: batch system software

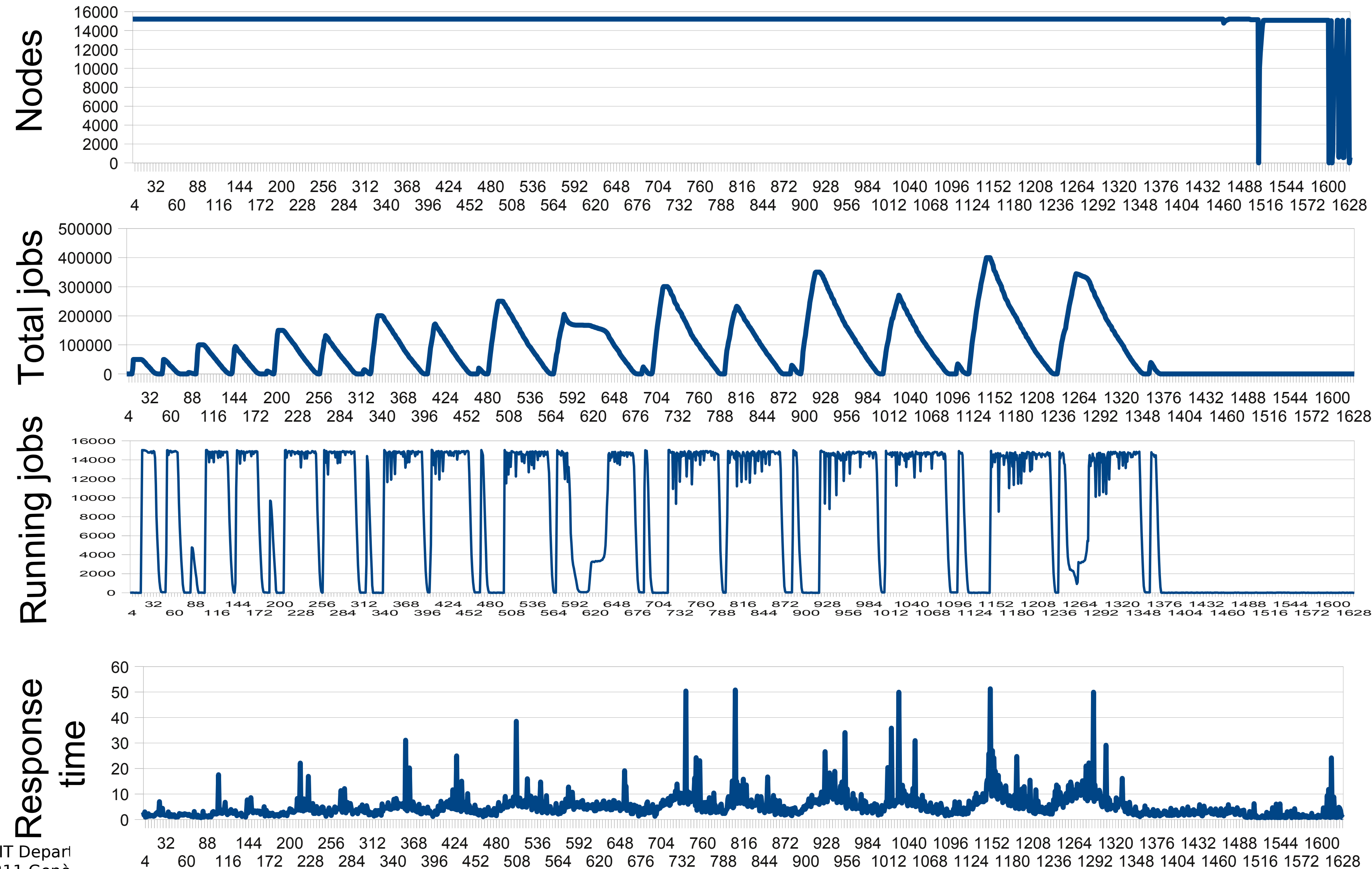
CERN IT Department

5000 worker nodes



CERN IT Department
CH-1211 Genève 23
Switzerland
www.cern.ch/it

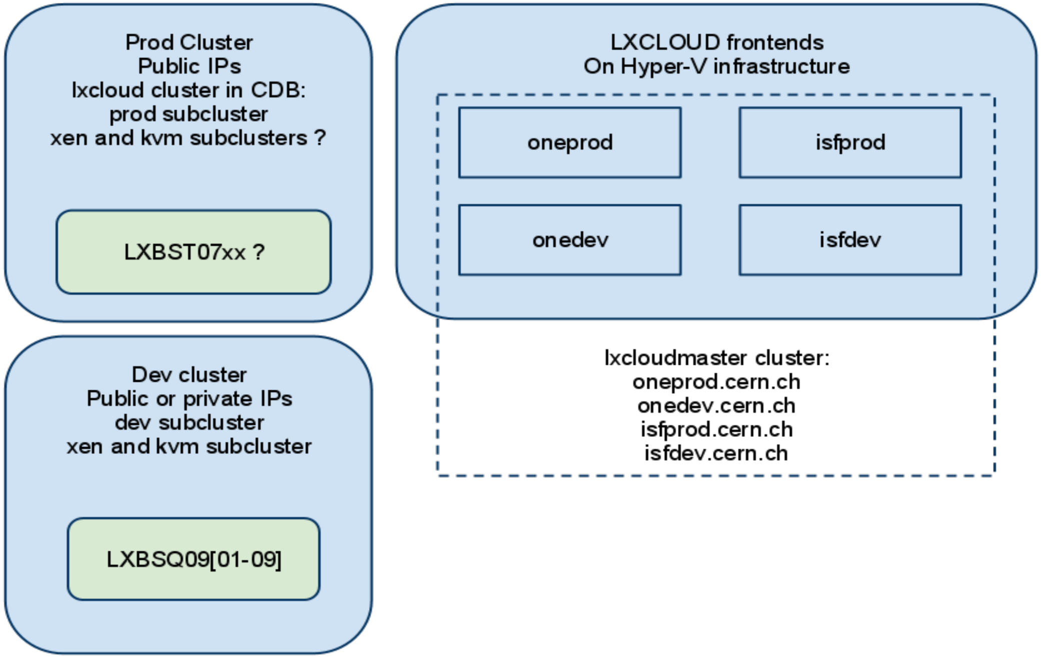
Scalability tests: batch system software



CERN IT Department
CH-1211 Genève 23
Switzerland
www.cern.ch/it

Future Plans

- Put a small number of virtual worker nodes in production in lxbatch
- Monitor performance, usage and overall processes
- Setup prod and dev machines for ONE and ISF and run both side by side



Future Plans and Technical issues:

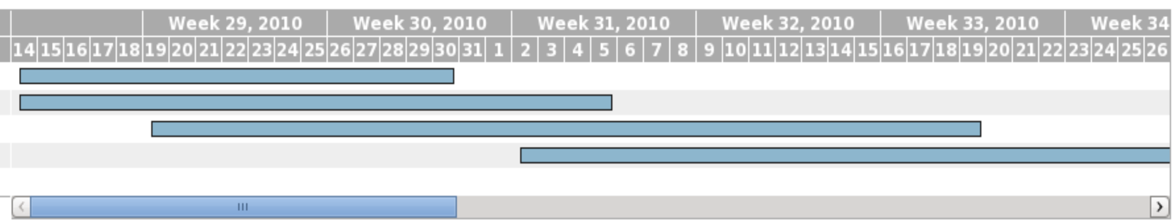
- VM CPU **accounting** and CPU factors
- Identify and solve possible storage issues (scratch, swap, AFS cache)
- Could have ~100 VM in prod lxbatch by ~autumn 2010, assuming no further surprises

LXCLOUD-Prod

Start: July 14, 2010
Finish: October 15, 2010
Report Date: July 14, 2010

Gantt Chart

WBS	Name	Work
1	Deploy lxcloud front-ends	13d
2	Quattorize ONE	17d
3	Develop VM monitoring solution	24d
4	Put VMs in Burn in lxbatch	24d
5	Put VMs in Prod lxbatch	30d



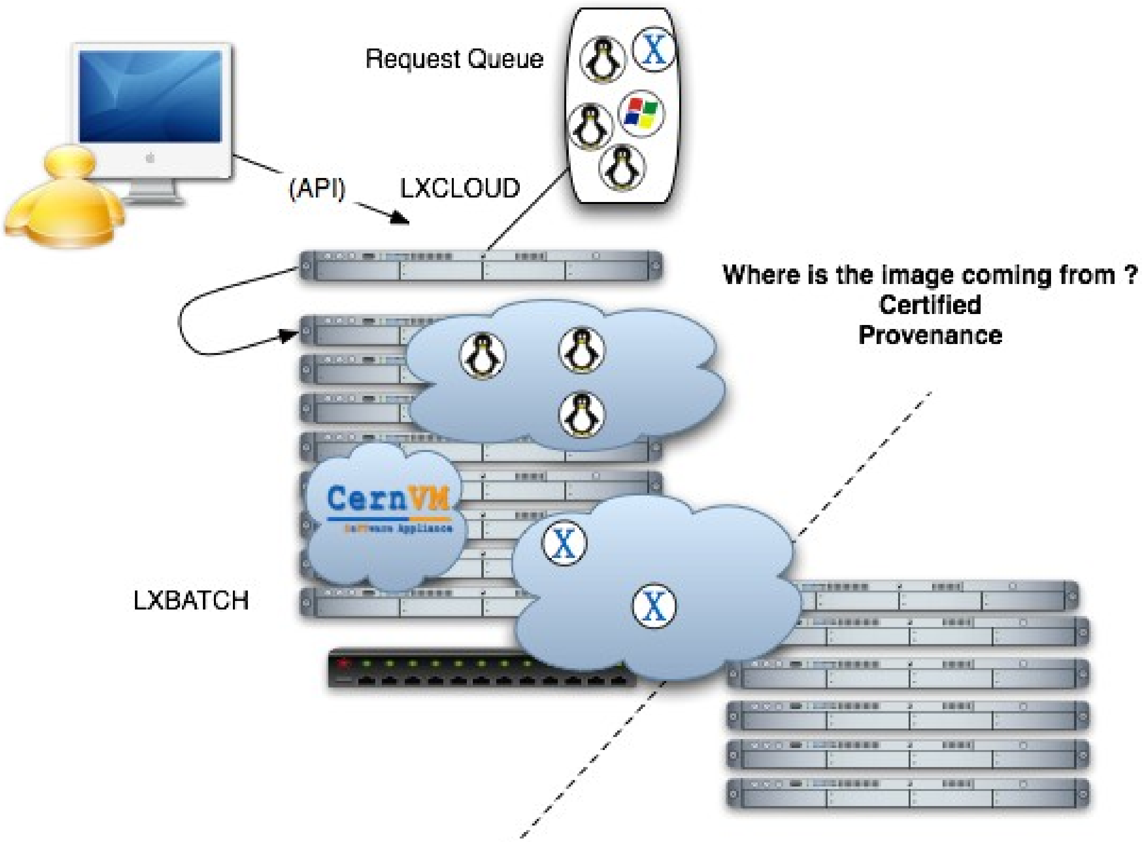
Tasks

WBS	Name	Start	Finish	Work	Priority	Complete	Cost
1	Deploy lxcloud front-ends	Jul 14	Jul 30	13d		0%	
2	Quattorize ONE	Jul 14	Aug 5	17d		0%	
3	Develop VM monitoring solution	Jul 19	Aug 19	24d		0%	
4	Put VMs in Burn in lxbatch	Aug 2	Sep 2	24d		0%	
5	Put VMs in Prod lxbatch	Sep 6	Oct 15	30d		0%	

Resources

Name	Short name	Type	Group	Email	Cost

Merging this effort with the work of HEPiX working group has great potential for CERN and WLCG.

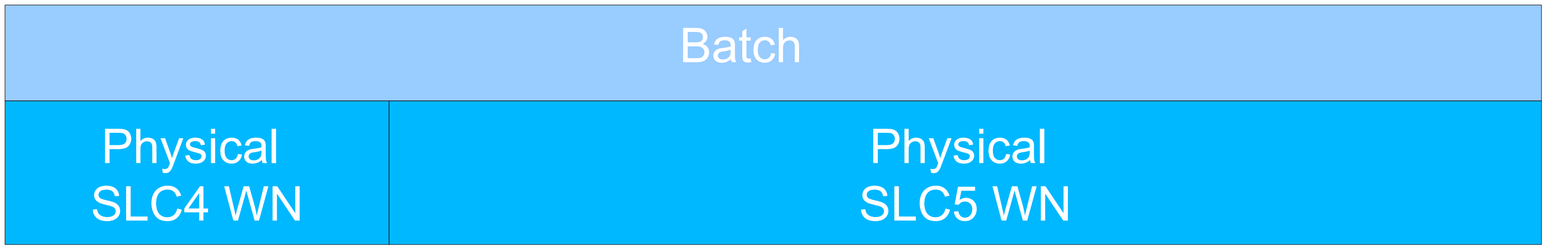


CERN IT Department
CH-1211 Genève 23
Switzerland
www.cern.ch/it

Long term ideas for CERN

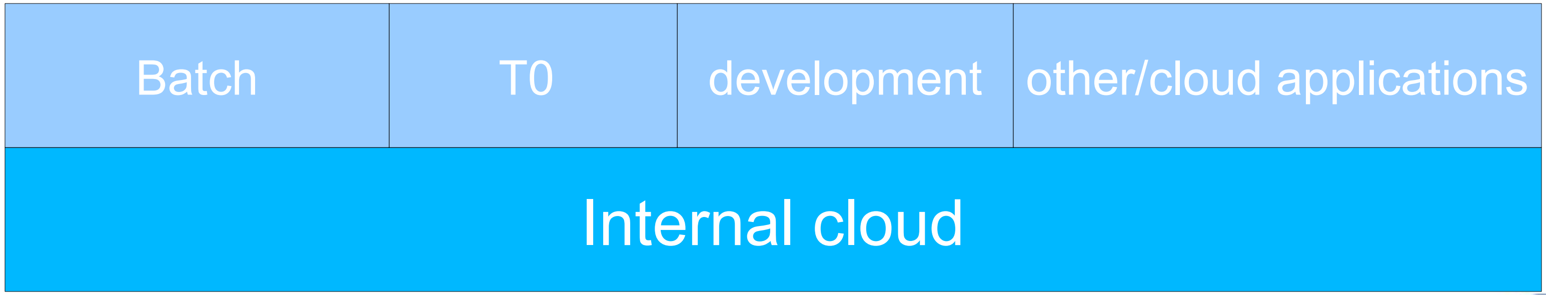
(SLC = Scientific Linux CERN)

Today



Near future:

(far) future ?



CERN IT Department
CH-1211 Genève 23
Switzerland
www.cern.ch/it

Conclusions

We have developed the mechanisms to offer an IaaS service on a large scale:

- We have a scalable hypervisor farm setup, supporting XEN and KVM
- We have an efficient image distribution system
- This IaaS provider currently moving to production for virtual worker nodes
- Technically able to provide IaaS at large scale
  - For IT
  - Maybe later also for users ?

Thank you!

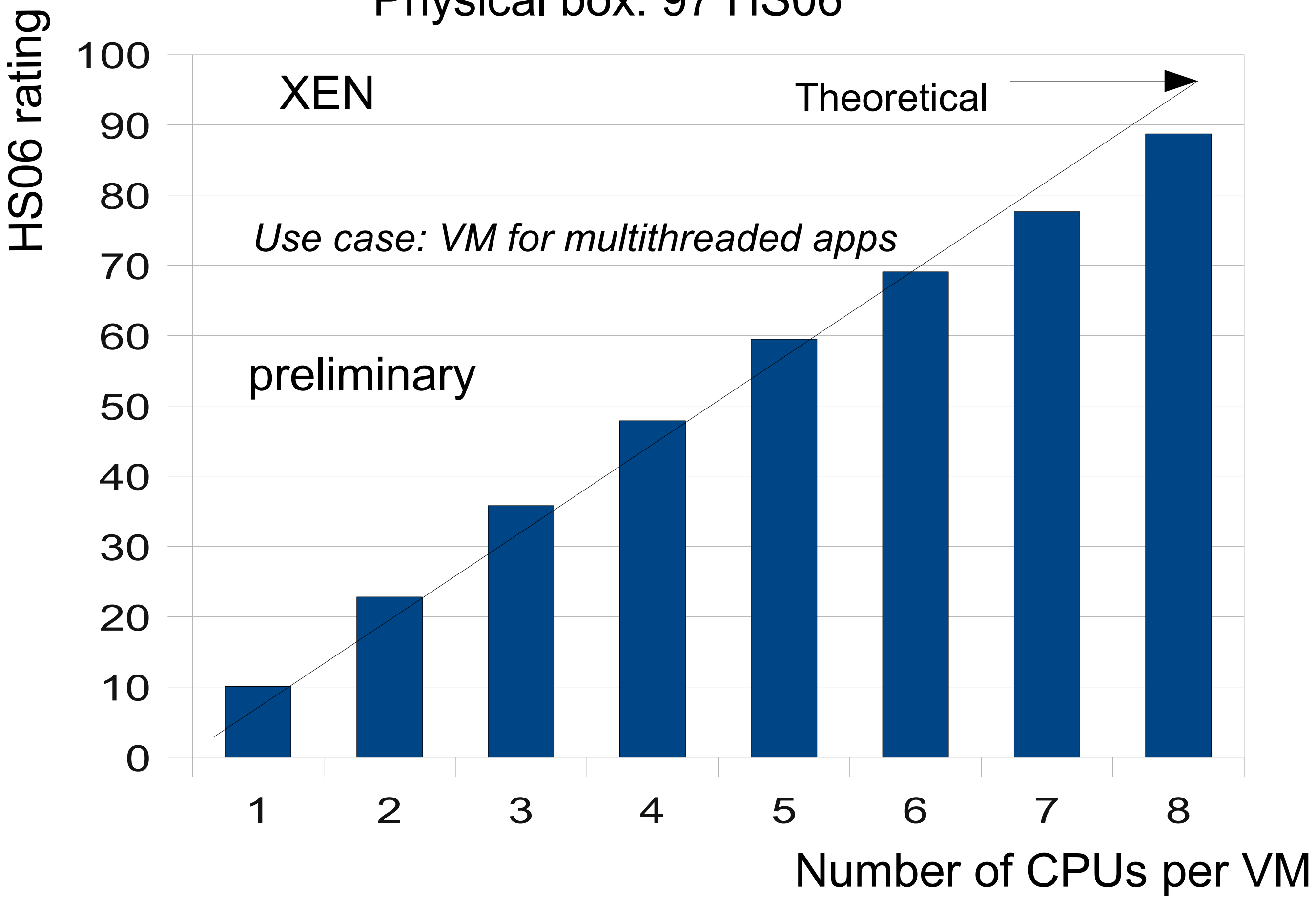
Any questions ?

CPU performance: HS06 benchmark results (preliminary)

CERN IT Department

Test case: 1 VM / hypervisor, raising the number of CPUs/hypervisor

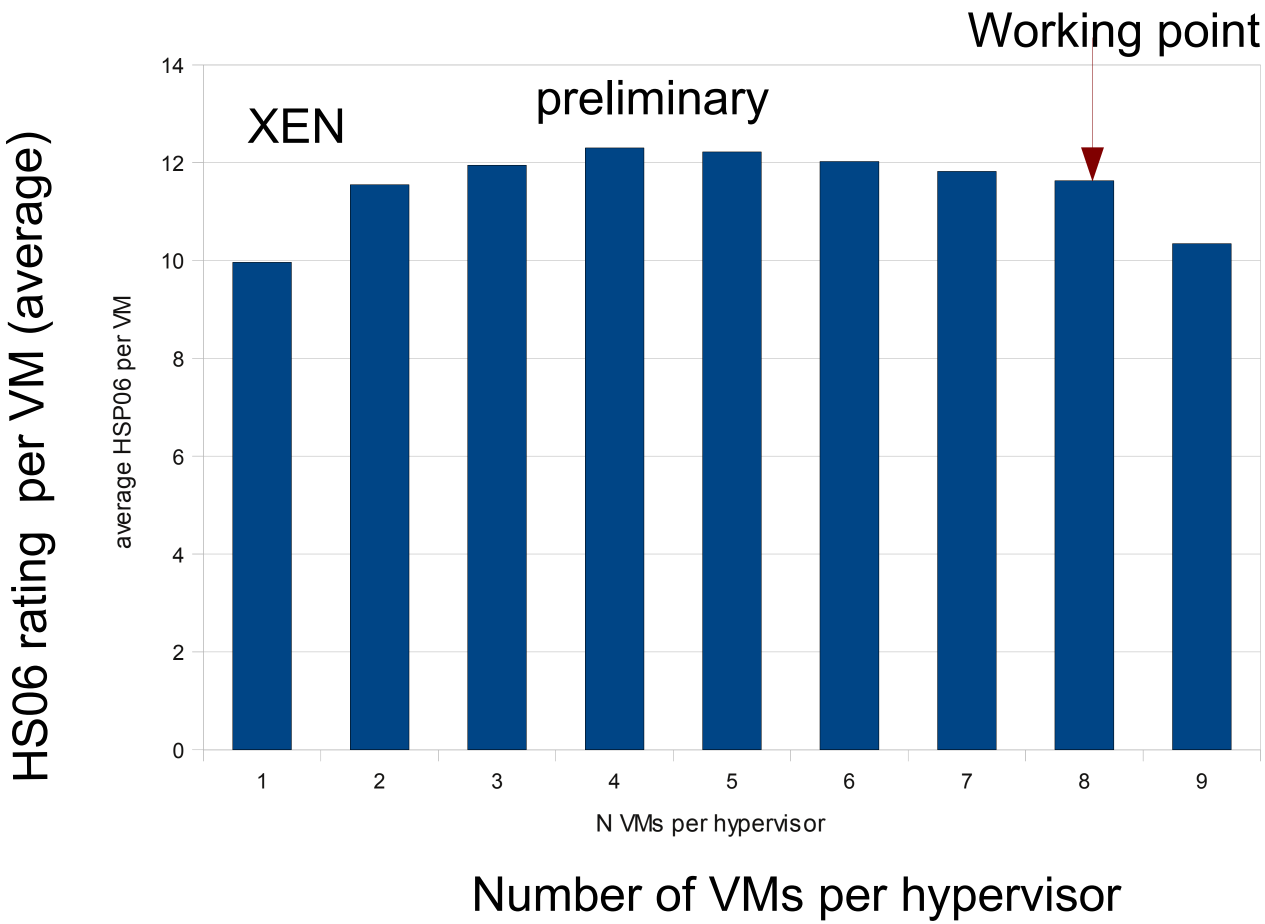
Physical box: 97 HS06



CPU performance: HS06 benchmark results (preliminary)

CERN IT Department

Test case : raise number of 1core VMs on a single hypervisor



CERN IT Department
CH-1211 Genève 23
Switzerland
www.cern.ch/it

Disk performance: iozone tests

Test case:

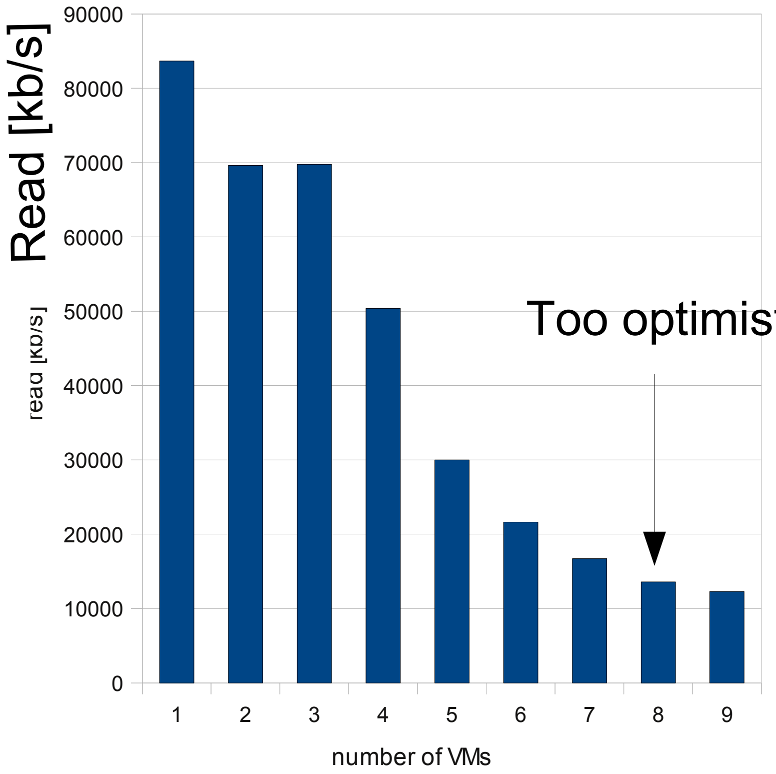
- Raise the number of VMs running iozone on a single hypervisor
- Compare with running N iozone processes in parallel on a single physical server

- Write performance OK
- Read performance requires tuning

Tuning: eg. change read-ahead size in
/sys/block/svdX/queue/read_ahead_kb

Command line parameters: iozone -Mce -r 256k -s 8g -f /root/iozone.dat -i0 -i1 -i

read performance



Compare with physical node:

~100MB/s read

~82MB/s write

~20% performance penalty

Needs further investigation !

Command line parameters: iozone -Mce -r 256k -s 8g -f /root/iozone.dat -i0 -i1 -i

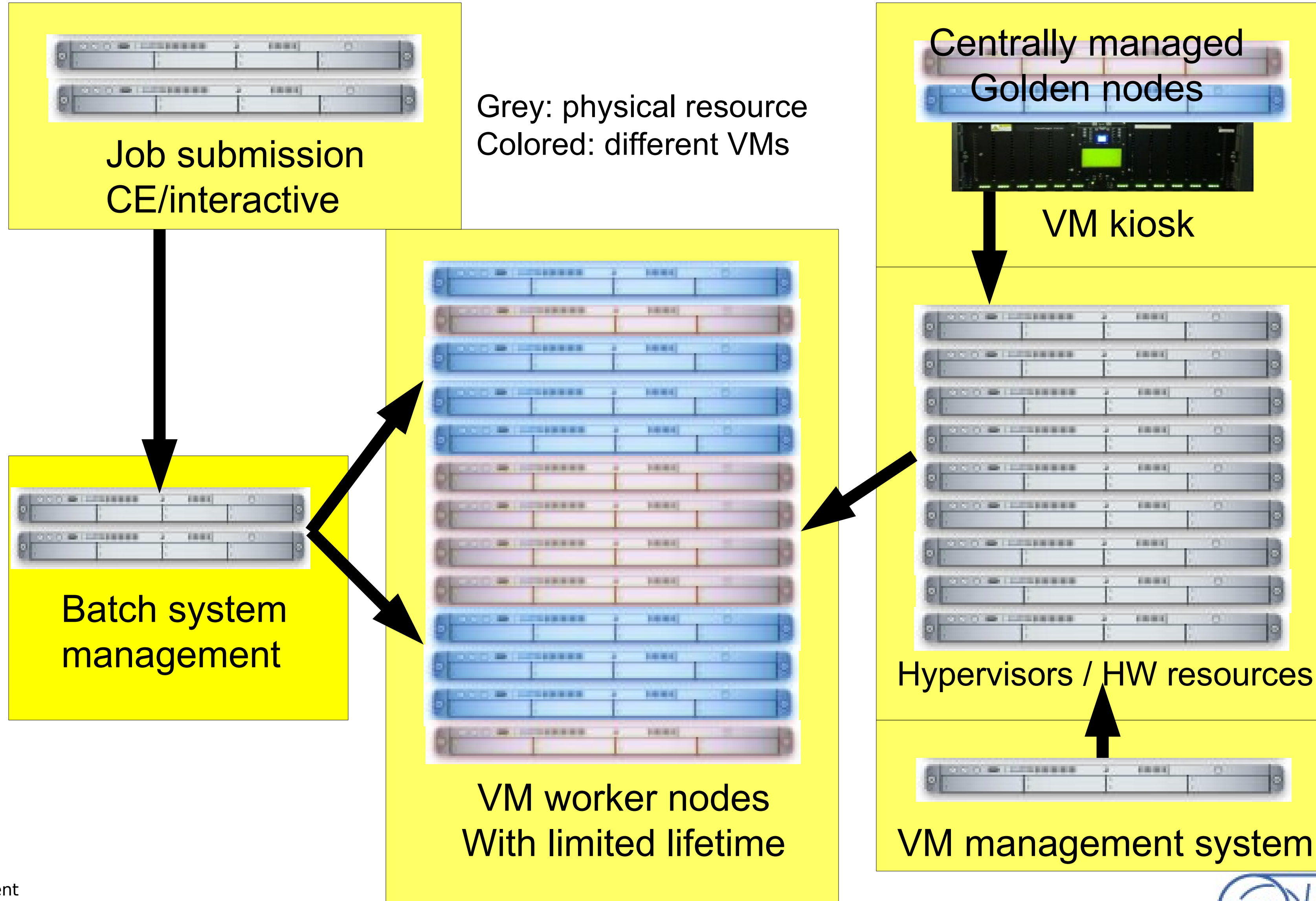
CERN IT Department
CH-1211 Genève 23
Switzerland
www.cern.ch/it

Network performance

Still to do ...

… no big worries here though

Batch virtualization: architecture



Cloud standards

No standards yet

- OGC OCCI, CCIF, Cloud Manifesto, Cloud Security Alliance

  ....lots of bodies at work (i.e according to NASA speaker ~20)
- deltacloud.org (Driven by RH, unifying API with adapters)
- Vcloud (from VMware) and EC2 API seem key for IaaS interface
- OpenNebula speaks Deltacloud and Deltacloud speaks OpenNebula

CLOUDCOMPUTINGINTEROPERABILITYFORUM